Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

Software Process Improvement: Toward a Comprehensive Framework for Research

Sirdhar Nerur
Texas A&M International University, nerur@tamiu.edu

Wullianallur Raghupathi
Californaia State University, rp@ecst.csuchino.edu

Software Process Improvement: Toward a Comprehensive Framework for Research

Sridhar Nerur, Dept. of Accounting and IS, Texas A&M International University, Laredo, TX 78041. (210) 326-2536. email: nerur@tamiu.edu
Wullianallur Raghupathi, Dept. of Accounting and MIS, California State University, Chico, CA 95929. (916) 898-4825. email: rp@ecst.csuchico.edu

Introduction

Software process improvement is rapidly becoming a broad area of research and application within the software engineering discipline. It is only recently that it has attracted the attention of researchers in the Information Systems field. This mini-track is evidence of that interest. The Information Systems field has important contributions to make in the understanding, development, and application of strategies and models in software process improvement. For example, the research on groups (GDSS) has a lot to contribute to the effective performance of teams in large software projects. Our understanding of end user computing (EUC) can provide valuable insights into the understanding of the people capability maturity model. Work in the decision support systems area can result in important collaboration in software risk assessment model development (Raghupathi, 1995). Research in business process redesign and reengineering can be applied to software configuration and change management. There is a parallel between Nolan's stages of growth (SOG) and the Software Engineering Institute's Capability Maturity Model (CMM). However, what is lacking is a framework to integrate the disparate research being conducted in several fields and sites and to offer new research topics for MIS researchers.

The purpose of this research paper is to develop a framework for research in software process improvement that researchers and practitioners in the MIS and related fields can use to identify and investigate topics of interest. The results of a factor analysis of author co-citation data showing tentative areas of research are presented. It is expected that this research endeavor will spawn research activities that will then lead to a cumulative body of literature that is both integrative and collaborative. Researchers from various disciplines can work together to examine various aspects of software process improvement. Such an effort will lead to a rather cohesive body of work - very different from the present disparate and ad hoc approach. Since the MIS field investigates information systems issues in an organizational context, it is important that researchers in the field participate in this momentous dialogue on software process improvement.

Software Process Improvement

Humphrey (1989, p. x) defines the software process as a "set of actions required to efficiently transform a user's need into an effective software solution." He further suggests that "many software organizations have trouble defining and controlling this process, which is where they have the greatest potential for improvement." A process that is both well-defined and well-managed and a consistent implementation are necessary to render the use of software technology in organizations effective. The problems that plague software development cannot be adequately addressed until the entire software task is viewed as a process that can be planned, controlled, measured, and ameliorated. This, according to Humphrey, is a critical initial step in resolving software problems. It is imperative that "a fully effective software process consider the relationships of all the required tasks, the tools and methods used, and the skill, training, and motivation of the people involved." (p. 4)

Humphrey (1989, p. 4) describes six steps that organizations must take to improve their software capabilities. It is in these six steps that many analogies to work in the MIS field can be drawn. The six steps are:

1. Get an understanding of the current state of software capabilities;

2. Develop a vision of the desired process;

3. Develop a prioritized list of necessary process improvement actions;

4. Formulate a plan to accomplish the necessary actions;

5. Dedicate the resources to operationalize and put the plan into effect; and

6. Start over at step 1.

John Rockart's CSFs approach can be used to identify and prioritize software process improvement actions. The six steps further breakdown into models, methodologies, assessments, and practices. For example, the Capability Maturity Model (CMM) characterizes the software process into one of five maturity levels, viz., initial, repeatable, defined, managed, and optimizing. This is used with an assessment methodology. Numerous other steps and models are described in Humphrey (1989, 1995). The CMM along with the assessment is now widely applied in industry. Similarly, the People Capability Maturity Model (P-CMM), a recently developed model, provides a maturity framework to focus on continuously improving, managing, and developing the human assets of a software or information systems organization (Curtis, Hefley and Miller, 1995). According to these researchers, in order to improve their performance, organizations must focus on three interrelated components - people, process, and technology.

Toward a Framework for Research

As indicated earlier, the primary purpose of this paper is to delineate a framework for research on various topics and issues in software process improvement. While the essence of the framework has been derived from the various activities of the Software Engineering Institute, an Author Cocitation Analysis (ACA) was done to ensure that the framework was conceptually consistent with the existing corpus of knowledge in this specialty. ACA, a form of bibliometrics, has been widely used to identify intellectual perspectives in a scientific specialty and also to derive the cognitive structure of a field of interest (McCain, 1990; Culnan, 1986; White and Griffith, 1981). ACA provides tremendous insights into the evolution of thought and the building of cumulative traditions within scientific fields. It is thus a viable research methodology for the development of a framework that will facilitate collaborative and meaningful research to enhance the scope and effectiveness of software process improvement.

There is a growing body of literature on software process improvement, software economics, risk management, estimation and assessment, software quality and reliability, software metrics, and so forth. A cursory survey of this literature yielded a list of 40 authors for our preliminary analysis. It is believed that these authors, through their seminal contributions to various aspects of software development, have laid a strong foundation for the construction of the edifice of software process improvement. The citation data for these authors was obtained from the Science Citation Index (SCI) for the period 1986-1995. The correlation matrix obtained from a matrix of raw cocitation counts between each pair of authors was then subjected to a statistical analysis using Multivariate techniques such as Factor Analysis and Multidimensional Scaling (MDS). This paper discusses only the results of the factor analysis. Each derived factor may be construed as a subfield or a research topic of interest. While MDS provides a map of authors in intellectual space, factor analysis reveals the breadth of an author's contributions (McCain, 1990). Table 1 shows the factors and the tentative names that have been attributed to each factor. Also listed under each factor are research areas that might be of interest to the MIS community. The seven factors accounted for 78.43% of the total variance. It can be easily seen that there is a lot of overlap between the factors, with some of the authors contributing to more than one factor.

Interdisciplinary Research Issues

In addition to the research topics provided by ACA, there are some interdisciplinary research issues to be expressly addressed by MIS researchers. A brief discussion of some of these follows.

Capability Maturity Model (CMM): The Software Engineering Institute (SEI) has started surveying user firms. The focus thus far has been on government contractors and predominantly software and hardware firms. Research opportunities in this area include the adaptation of the CMM for businesses and industries such as banking, manufacturing, service (egs., airlines, hotels, etc.,), and small businesses. Further, both quantitative and qualitative models have to be developed to assess the pros and cons and to weigh the costs and benefits of undertaking a software process improvement activity. Experiences from outsourcing studies can be applied in this area.

People Capability Maturity Model: This is a relatively recent model - there is potential to study teams using research findings in GDSS, and also to study the impact of information technology on team learning (learning in organizations - high/low IT vs. high/low learning).

Curriculum Development: There is an urgent need for a software process improvement course that is integrative, i.e., it should draw on theories, models, and methods from computer science, management science, and information systems. For MIS departments to include this all important course in the MIS curriculum, we need to develop alternative models of courses focusing on issues in software process improvement. How to improve the Business Software Process would be the critical question in this course.

Organization Capability Development (OCD): The purpose of this project at SEI is to facilitate transition of new technologies into routine use in software organizations, to build organizational infrastructures to foster continuous improvement and transition SEI skills into client organizations, and so forth (Bridge, Issue 3, 1994, p. 1). Potential research issues are discussed in that article.

Multimedia Technology in Software Development: Research in CASE tools and the impact of CASE tools on the systems development process can give insights into the potential benefits of multimedia technology.

Software Risk Assessment: The potential of Decision Support Systems in software risk assessment has been investigated (Raghupathi, 1995). This area opens up a large number of research issues including development,

Table 1: Author Factor Loadings 0.4

Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7
Yourdon .91 Demarco .85 Freeman .73 Parnas .72 Boehm .58 Brooks .67 Charette .44 Glass .43	Kemerer .91 Banker .78 Albrecht .77 Abdel-Hamid .56 Conte .6 Mukhopadhayay .65 Putnam .73 *Software Project Management/Measurem*	McCabe .95 Halstead .95 Shen .85 Myers .64 Basili .75 Albrecht .51 Boehm .49	Cooper .96 Jones .9 Carr .8 Glass .66 Mukhopadhayay .44 *Software Quality* Other research areas:	Humphrey .9 Paulk .84 Grady .62 Cusumano .59 Pfleeger .56 Charette .54	Musa .93 Littlewood .93 Ramamoorthy .62 Mills .44 Fagan .41 Leveson 0.4 * *Software*	Neumann .91 Leveson .85 Parnas .52 *Software Risk/Safety Management* Other research areas: Human

McFarlan .62 Mills .54 Myers .55 Pressman .72 Ramamoorthy .51 *Software Life-Cycle Management* Other research areas: process models, software economics, methodology issues, design automation, project management, *peopleware*	*ents* Other research areas: models for estimation , finance models, software reuse, software metrics	Card .71 Conte .63 Curtis .77 Gilb .58 Grady .42 *Software Metrics* Other research areas: Software design complexity, project complexity, effort, productivity, and quality, software science	Software risk management, process improvement, software maintenance	Fagan .47 Basili .44 Boehm .41 Brooks .41 Card .4 *Software Process Improvement* Other research areas: managing software, software engineering standards, risk management, quality and productivity	*Reliability* Other research areas: Software inspections, error reduction, quality, productivity * Leveson had a loading of .399	safety, fault tolerance, security

implementation, validation, and use of DSS for risk assessment.

Software measurement (Jones, 1991), software technology transition, configuration management, software sizing and estimating, software quality, and CMM vs. ISO 9000 series comparisons are but a few of the other potential research areas.

Conclusions

In this paper, we have outlined a preliminary framework based on an Author Cocitation Analysis. These are exciting times for the MIS field. There are ample research opportunities in the software process improvement area. While we continue to look into development methodologies, outsourcing, EUC, impact of IT on organizations, measuring the value of IS, GDSS and other topics, the quest for software process improvement opens up whole new areas of research for the MIS community well into the 21st century. There is an urgent need for more empirical studies in the research topics discussed above. It is up to us to form coalitions and collaborate to develop and investigate these new ideas.

The proposed framework can be used to develop individual research agendas in the software process improvement area. In order for research to generate cumulative knowledge, a common perspective regarding the purpose and characteristics of these topics is required. This framework can be extended to

include taxonomies outlining models, methods, applications, and variables critical to software process improvement that identify research questions and organize research results.

References available upon request from Sridhar Nerur.